

Plot Extracted and Derived Data

Â

[Contents](#) [Previous](#) [Next](#)

Goal: Guide you through some basic features of extracting and scripting derived data.

Before running the tutorial below, type *"python"* or *"cdat"* at the command line.Â You will see the python prompt appear (i.e., ">>>"). You can now enter the command lines below.

You can now enter the command lines below.

You can [view](#)Â or [download](#)Â the full source code. To run the source code at the command line, type: *"python extract_and_plot.py"*.

```
# Import the modules needed for the tutorial
# cdms - Climate Data Management system accesses gridded data.
# vcs - Visualization and control System 1D and 2D plotting routines.
# cdutil - Climate utilizes that contains miscellaneous routines for
#           manipulating variables.
# time - This module provides various functions to mainpulate time values.
# os - Operation System routines for Mac, DOS, NT, or Posix depending on
#      the system you're on.
# sys - This module provides access to some objects used or maintained by
#      the interpreter and to functions that interact strongly with the interpreter.
import vcs, cdms, cdutil, time, os, sys

# Open data file:
filepath = os.path.join(sys.prefix, 'sample_data/ts_da.nc')
cdmsfile = cdms.open( filepath )

# Extract a 3 dimensional data set
data = cdmsfile('ts')

# From viewing the dataset's attributes, we see that
# it is "Surface Air Temperature" and its units are
# represented in kelvin (K).
data.info()

# Initial VCS:
v = vcs.init()

# Plot data using the default boxfill graphics method:
v.plot( data )

# Select one time step, and average over the longitude
# axis resulting in a zonal mean
dl=cdutil.averager(data(time=7665, squeeze=1), axis='x')

# Set the variable's ID to 't_z'.
dl.id = 't_z'

# Clear the VCS Canvas and plot the 1D dataset.
v.clear()
v.plot( dl )

# Subtract 273.16 to produce temperature in degrees C
dc = data - 273.16
```

```

dc.id = 'ts'
dc.long_name = 'Surface (2m) Air Temperature [C]'
v.clear()
v.plot( dc )

# Extract a 4 dimensional dataset
filepath = os.path.join(sys.prefix, 'sample_data/ta_ncep_87-6-88-4.nc')
cdmsfile = cdms.open( filepath )
data = cdmsfile('ta')

# Average over time and longitude to get a variable
# with latitude and level axes
d2 = cdutil.averager(data, axis='tx')
d2.id = 't_zh'

# Plot results
v.clear()
v.plot( d2 )

# Extract data from a specific level
dp = cdmsfile('ta', longitude=(180, -180), latitude = (90., -90.), level =(200., 200.), squeeze=1

# Plot results
v.clear()
v.plot( dp )

```

Â

[Contents](#) [Previous](#) [Next](#)